基于 Im2col 的并行深度卷积神经网络优化算法 *

胡　健 [1,2]，龚　克 [1]，毛伊敏 [1†]，陈志刚 [3]，陈　亮 [2]

(1. 江西理工大学 信息工程学院，江西 赣州 341000; 2. 赣南科技学院 电子信息工程学院，江西 赣州 341000; 3. 中南大学 计算机学院，长沙 410083)

摘　要：针对大数据环境下并行深度卷积神经网络(DCNN)算法中存在数据冗余特征多、卷积层运算速度慢、损失函数收敛性差等问题，提出了一种基于 Im2col 方法的并行深度卷积神经网络优化算法 IA-PDCNNOA。首先，提出基于 Marr-Hildreth 算子的并行特征提取策略 MHO-PFES，提取数据中的目标特征作为卷积神经网络的输入，有效避免数据冗余特征多的问题；其次，设计基于 Im2col 方法的并行模型训练策略 IM-PMTS，通过设计马氏距离中心值去除冗余卷积核，并结合 MapReduce 和 Im2col 方法并行训练模型，提高了卷积层运算速度；最后，提出改进的小批量梯度下降策略 IM-BGDS，排除异常节点的训练数据对批梯度的影响，解决了损失函数收敛性差的问题。实验结果表明，IA- PDCNNOA 算法在大数据环境下进行深度卷积神经网络计算具有较好的性能表现，适用于大规模数据集的并行化深度卷积神经网络模型训练。

关键词：大数据；DCNN 算法；并行计算；特征提取；图像分类

中图分类号：TP311　　doi: 10.19734/j.issn.1001-3695.2022.03.0114

Parallel deep convolution neural network optimization based on Im2col

Hu Jian[1, 2], Gong Ke[1], Mao Yimin[1†], Chen Zhigang[3], Chen Liang[2]

(1.*School of Information Engineering*, *Jiangxi University of Science & Technology*, *Ganzhou Jiangxi* 341000, *China*; 2.*Electronic Information Engineering*, *Gannan University of Science & Technology*, *Ganzhou Jiangxi* 341000, *China*; 3.*College of Computer Science & Engineering*, *Central South University*, *Changsha* 410083, *China*)

Abstract: In the large data environment, there are many problems in the parallel deep convolution neural network (DCNN) algorithm, such as excessive data redundancy, slow convolution layer operation and poor convergence of loss function. This paper proposed a parallel deep convolution neural network optimization algorithm based on the Im2col method. First, the algorithm proposed a parallel feature extraction strategy based on Marr-Hildreth operator to extract target features from data as input of convolution neural network, which can effectively avoid the problem of excessive data redundancy. Secondly, the algorithm designed a parallel model training strategy based on the Im2col method. The redundant convolution kernel is removed by designing the Mahalanobis distance center value, and the convolution layer operation speed is improved by combining the MapReduce and Im2col methods. Finally, the algorithm proposed an improved small-batch gradient descent strategy, which eliminates the effect of abnormal data on the batch gradient and solves the problem of poor convergence of the loss function. The experimental results show that IA-PDCNNOA algorithm performs well in deep convolution neural network calculation under large data environment and is suitable for parallel DCNN model training of large datasets.

Key words: big data; DCNN algorithm; parallel computing; feature extraction; image classification

0　引言

DCNN[1]作为深度学习领域中一类重要的分类算法，具有强大的表征能力、泛化能力和拟合能力，效果稳定且无须对数据做额外的特征工程，常被运用于图像分类[2]、语音识别[3]、对象检测[4]、语义分割[5]、人脸识别[6]、自动驾驶[7]等领域，受到人们的广泛关注和深入研究。

近年来，随着移动互联网的发展以及数据存储介质容量的突破，产生了海量的、多模态的、高价值的数据[8]，众多科研者和公司尝试从中提取高价值的信息，但海量的数据使得 DCNN 模型的训练将面临大量时间消耗，数据与模态变化又将导致模型参数需要反复训练等困难。因此，如何降低大数据环境下 DCNN 模型训练的代价成为了一个亟待解决的问题。

Google 公司开发的 MapReduce 并行计算模型以其易于编程、高容错性、均衡负载和扩展性强等优点深受广大学者和企业的青睐，许多基于 MapReduce 计算模型的 DCNN 算法也得到了广泛的研究[9~12]。文献[13]提出基于 MapReduce 的并行化 DCNN 算法，该算法采用分而治之的思想，通过 MapReduce 的 Split 方法对数据进行划分，构建多个计算节点同时训练 DCNN 网络模型，选取准确率最高的网络模型作为算法的输出，实现了 DCNN 并行化训练过程。基于此，文献[14]提出并行深度卷积神经网络算法 FCNN (Fully CNN for processing CT scan image)，算法将全视图转变为稀疏视图，并通过高斯滤波器，对特征边缘进行平滑处理，增强重要的纹理特征信息。虽然算法在将全视图转变为稀疏视图的过程会加快读取速度，但由于稀疏视图的特征结构变化，导致其难以对特征进行筛选，使得模型在训练的过程中会存在

收稿日期：2022-03-20；**修回日期：**2022-05-13　**基金项目：**科技创新 2030-“新一代人工智能”重大项目(2020AAA0109605)；国家自然科学基金资助项目(41562019)；江西省教育厅科技项目(GJJ209405，GJJ209406，GJJ209407)

作者简介：胡健(1967-)，男，江西赣州人，教授，硕导，博士，主要研究方向为数据挖掘和人工智能；龚克(1997-)，男，江西南昌人，硕士，主要研究方向为数据挖掘和人工智能；毛伊敏(1970-)，女(通信作者)，江西赣州人，教授，硕导，博士，主要研究方向为分布式计算和人工智能(mymlyc@163.com)；陈志刚(1960-)，男，湖南长沙人，教授，博导，博士，主要研究方向为分布式计算和人工智能；陈亮(1982-)，男，江西赣州人，讲师，硕士，主要研究方向为数据挖掘和人工智能.

数据冗余特征多的问题。文献[15]基于 Im2col 方法，提出单跨步优化 CNN 算法 SSOCNN (An optimization of im2col, an important method of CNNs based on continuous address access)，该算法设计基于连续内存地址读取的单跨步情况下的 im2col 算法加速方法，通过改变数据读取顺序，加速图像映射成矩阵的进程，并利用通用矩阵乘法对列向量和卷积核进行矩阵相乘运算，实现了对卷积层运算的加速，其本质是一种模型并行方法。但在构建并行卷积运算的过程中，算法难以筛除分散在各个节点的冗余卷积核，导致在大数据环境下，无法解决卷积层运算速度慢的问题。文献[16]通过将 DCNN 与萤火虫算法相结合，提出 MR-FPDCNN 算法 (Deep convolutional neural network algorithm based on feature graph and parallel computing entropy using MapReduce)，该算法将信息共享搜索策略与萤火虫算法相结合来寻找网络模型最优参数，并通过 MapReduce 通信机制共享 DCNN 网络参数，加快了损失函数的收敛速度，其本质是一种数据并行方法。但该算法没有针对异常节点的训练数据进行甄别处理，使得算法在反向传播过程中的损失函数收敛震荡，导致损失函数收敛性差。

综上，尽管上述算法取得一定的成效，但对于数据冗余特征多、卷积层运算速度慢、损失函数收敛性差等问题仍然是目前亟待解决的。针对以上问题，本文在 MapReduce 并行计算框架的基础上，提出一种基于 Im2col 算法的并行深度卷积神经网络优化算法 IA-PDCNNOA，算法在三个方面对并行深度卷积神经网络进行优化：a)在特征并行提取阶段，对于数据的预处理，现有算法大都是去除噪声与缺失值，进行数据标准化与正则化，但在大数据环境下，常规的数据预处理操作不仅难以提升模型精度，而且不能降低大数据所带来的计算资源消耗。IA-PDCNNOA 算法改进了 Marr-Hildreth 算子提出 MHO-PFES 策略，通过提取图像数据的全部边缘特征，评估同类型数据的特征相似度，并删除低相似度的特征来解决数据冗余特征多的问题，不仅去除无关特征对模型训练的精度影响，而且降低了计算资源的开销。b)在模型并行训练阶段，对于模型的卷积运算，算法提出基于数据并行的 IM-PMTS 策略，设计了马氏距离中心值 $MDCV$ 来去除当前卷积层中的冗余卷积核，并结合 Im2col 算法，将卷积运算的过程转换为矩阵运算，进而使得这一过程能够与 MapReduce 结合，进行并行化运算来提升卷积层的运算速度。c)在参数并行更新阶段，对于反向传播过程，大多并行算法采用随机梯度下降法或批梯度下降法进行参数的更新，然而，他们没有考虑到计算节点由于程序中断、内存溢出等因素而导致训练数据异常，这些异常数据会使得反向传播过程中的损失函数收敛震荡，进而导致损失函数收敛性差。IA-PDCNNOA 算法基于模型并行提出 IM-BGDS 策略，通过评估每个节点的梯度与批梯度间的距离，来去除批梯度中的异常值，实现批梯度的自适应调整，解决了损失函数收敛性差的问题。

综上所述，本文主要贡献包括：a)提出 MHO-PFES 策略，提取数据中的目标特征作为卷积神经网络的输入，解决了数据冗余特征多的问题；b)提出 IM-PMTS 策略，通过设计马氏距离中心值 $MDCV$ 来去除冗余卷积核，并结合 MapReduce 和 Im2col 方法并行训练模型，提高了卷积层运算速度；c)提出 IM-BGDS 策略，排除异常节点的训练数据对批梯度的影响，解决了损失函数收敛性差的问题。

1 相关概念介绍

定义 1 非局部均值算法[17]。非局部均值算法是一种基于邻域像素的滤波方法，可以结合图像中的全局信息进行数据降噪。假设 g 表示数据样本，$x,y\in g$，x 表示搜索窗口，y 表示邻域窗口，则降噪后灰度值 $\tilde{u}(x)$ 为可以表示为

$$\tilde{u}(x)=\sum_{y\in G}\phi(x,y)*\theta(y) \tag{1}$$

其中，$\phi(x,y)$ 为 x 与 y 区域间的相似度，$\theta(y)$ 为含噪声图像。

定义 2 余弦相似度[18]. 余弦相似度是度量个体之间的相似性指标。可以将个体的指标数据映射到向量空间，并测量两个个体向量之间的内积空间夹角余弦值，从而比较个体相似度。假设 X,Y 表示对比个体，$\vec{x},\vec{y}$ 表示个体 X,Y 的一维形式，则余弦相似度 $Sim(X,Y)$ 可以表示为

$$Sim(X,Y)=cos\theta=\frac{\vec{x}\cdot\vec{y}}{\|\vec{x}\|+\|\vec{y}\|} \tag{2}$$

其中，$\|\vec{x}\|,\|\vec{y}\|$ 为个体的模。

定义 3 Image to column, Im2col[19]。Im2col 是一种将卷积计算变换成矩阵乘法计算的变换函数。可以将输入值的 3D Matrix 转换为 2D Matrix，并将卷积核转换为 1D Matrix，从而实现将卷积计算转换成矩阵相乘计算。假设 x 表示输入特征图，w 表示卷积核，i,j 表示像素点横纵坐标，则卷积结果 $a_{i,j}$ 可以表示为

$$a_{i,j}=\sum_{h=0}^{kH-1}\sum_{w=0}^{kW-1}w_{h,w}x_{i+h,j+w} \tag{3}$$

其中，kH,kW 为款集合的宽和高。

定义 4 马氏距离[20]。马氏距离是度量学习中一种常用的距离指标。它是一种有效的计算两个未知样本集的相似度的方法。假设 S 表示多维随机变量的协方差矩阵，μ 表示样本均值，x 表示当前数据点。则马氏距离 $D_M(x)$ 可以表示为

$$D_M(x)=\sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{4}$$

2 IA-PDCNNOA 算法

IA-PDCNNOA 算法主要包括三个阶段：特征并行提取，模型并行训练，参数并行更新。a)特征并行提取阶段：提出 MHO-PFES 策略，首先通过改进 Marr-Hildreth 算子提取原始数据集的数据特征，然后筛选数据的目标特征作为卷积神经网络的输入，从而解决了数据冗余特征多的问题；b)模型并行训练阶段：提出 IM-PMTS 策略，首先设计马氏距离中心值 $MDCV$ 对同层卷积核剪枝，然后通过结合 MapReduce 和 Im2col 方法并行训练的方式加速卷积运算的过程，提高了卷积层运算速度；c)参数并行更新阶段：提出 IM-BGDS 策略，首先设计损失求和梯度 $LSG(T)$ 构建小批量数据梯度，然后通过误差反向传播算法对参数并行更新，排除异常节点的训练数据对批梯度的影响，解决了损失函数收敛性差的问题。

2.1 特征并行提取

目前在大数据环境下的并行 DCNN 算法中，初始图像数据中存在大量冗余特征，这些冗余特征未能经过有效筛选，使得在模型训练过程中存在数据冗余特征多的问题。为了解决此问题，提出了基于 Marr-Hildreth 算子的 MHO-PFES 策略，该策略主要包含两个步骤：a)特征提取：提出改进的非局部均值滤波器 $FT(\vec{a},\vec{b})$ (Filter transformation)对输入的图像数据进行滤波，并计算滤波数据的拉普拉斯方程 $h(x,y)$，寻找拉普拉斯方程的零交叉来提取图像特征；b)特征筛选：为进一步筛选目标特征，提出特征相关指数 $FCI(x,y)$ (Feature correlation indices)对比任意两个图像块间的相似度，并设定相关性系数 ε，通过去除 $FCI(x,y)<\varepsilon$ 的图像块来减少数据中的冗余特征。

1)特征提取

为了获取到高精度的图像特征，需先对初始数据集进行噪声去除，因此提出基于余弦相似度的非局部均值滤波器 $FT(\vec{a},\vec{b})$，通过图像在不同区域的自相似性来去除数据噪声；

然后再通过卷积核 $f(x,y)$ 与图像 $g(x,y)$ 的拉普拉斯运算，构建并寻找拉普拉斯方程的零交叉来提取数据特征，其具体过程为：首先，在目标图像设置以像素点 a 为中心的邻域窗口矩阵与以像素点 b 为中心的搜索窗口矩阵，使邻域窗口在当前图像中进行滑动，通过对比像素点 a,b 所在矩阵的余弦相似度得到邻域窗口的加权值，并根据权重值以及各个点本身的灰度值对数据进行降噪处理，得到降噪后图像 $g(x,y)$；接着，设置大小为 3*3 的卷积核 $f(x,y)$，对 $g(x,y)$ 进行拉普拉斯运算，得到拉普拉斯方程 $h(x,y)=\nabla^2(f(x,y)\cdot FT(\vec{a},\vec{b}))$；最后，判断当前节点的拉普拉斯方程的二阶导数是否为交叉零点，且此节点的一阶导数处在较大峰值，若满足条件则将此节点保留，否则将此像素点置零，然后合并当前数据节点得到特征提取后的图像。

定理 1　基于余弦相似度的非局部均值滤波器 $FT(\vec{a},\vec{b})$）。已知 $\vec{a},\vec{b}$ 分别表示以像素点 a 为中心的邻域窗口矩阵与以像素点 b 为中心的搜索窗口矩阵。变换函数 $FT(\vec{a},\vec{b})$ 的计算公式如下：

$$FT(\vec{a},\vec{b})=\sum_{b\in G_i}\frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|+\|\vec{b}\|}\cdot\theta(a) \tag{5}$$

其中 θ 为含噪声图像，G_i 为当前图像数据。

证明　非局部均值滤波原理利用了噪声的非相关性特征，设无噪声的像素块的值为 $\omega(x,y)$，噪声值为 $\psi(x,y)$，则与噪声融合后的像素块的值为 $\rho(x,y)=\omega(x,y)+\psi(x,y)$，相似像素块叠加后取均值得到 $\bar{\rho}(x,y)=1/k\cdot\sum_{i=1}^{k}\rho_i(x,y)$，则 $\bar{\rho}(x,y)$ 的期望为 $E[\bar{\rho}(x,y)]=1/k\cdot\sum_{i=1}^{k}(E[\omega_i(x,y)]+E[\psi(x,y)])$。由于像素块的相似性，$E[\omega_i(x,y)]$ 可简化为 $\omega(x,y)$，当噪声为 0 时，$E[\psi(x,y)]=0$，故 $E[\bar{\rho}(x,y)]=\omega(x,y)$。此外，由于噪声的非相关性，$\omega(x,y)$ 的方差为 $\sigma[\bar{\rho}(x,y)]^2=\sigma[\omega(x,y)]^2+1/k^2(\sum_{i=1}^{k}\sigma_{\psi_i}^2)^2$，由于 $\omega(x,y)$ 无噪声，方差为 0，故 $\sigma[\bar{\rho}(x,y)]^2=1/k\cdot\sigma[\psi(x,y)]^2$，则表明噪声 $\psi(x,y)$ 与方差相关，$FT(a,b)$ 通过减小 $\psi(x,y)$ 来降低数据噪声。证毕。

2)特征筛选

在完成特征提取后，策略将 $batch$ 中图像切块，并提出特征相关指数 $FCI(x,y)$ 来计算任意两个图像块之间的特征相似度，然后去除 $FCI(x,y)<\varepsilon$ 的图像块来实现数据中冗余特征的去除，具体过程如下：首先，将相同类别的图像切分至等大小的图像块，并提出特征相关指数 $FCI(x,y)$ 来计算任意两个图像块之间的相似度，其中 x,y 表示两个互不相同图像块；接着，映射键值对 $<(x,y),FCI(x,y)>$ 存储至 HDFS 中，设定并根据相关性系数 ε 去除键值对中 $FCI(x,y)<\varepsilon$ 的项，减少图像中的冗余特征；最后，再次遍历键值对，读取 HDFS 中剩余键值对的 *key* 来获取冗余特征筛选后图像块的编号，并将筛选后的图像块作为卷积神经网络的输入，完成数据的特征筛选。

定理 2　特征相关指数 $FCI(x,y)$．已知 x 和 y 分别表示两条特征向量，μ_x,μ_y 表示 x 和 y 的期望，σ_x,σ_y 表示 x 和 y 的方差。特征相关指数 $FCI(x,y)$ 的计算公式为

$$FCI(x,y)=\frac{2\sigma_x\cdot\sigma_y}{\sigma_x^2+\sigma_y^2+(\mu_x-\mu_y)^2} \tag{6}$$

证明　$FCI(x,y)$ 是衡量 x 和 y 之间的特征相似度的指标，设 μ_x,μ_y 表示 x 和 y 的期望，σ_x,σ_y 为 x 和 y 的方差，当特征向量 x 在 $\sigma_x=0$ 时，卷积过程在 x 上的操作属于线性叠加，无法对特征进行抽取，此时 $FCI(x,y)=0$；当 $\sigma_x\neq0,\sigma_y\neq0$ 且特征向量 x 和 y 的特征相似时，$FCI(x,y)\to1$。证毕。

算法 1　征并行提取算法

输入：　批数据 $batch$，超参数 ε

输出：　特征提取后数据 $batch$

a)　RunMapRedece（ $batch,\varepsilon$ ）

b)　For each g_i in $batch$ do

c)　　MapReduce.Map（ g_i ）

d)　　　For each x_j in g_i do

e)　　　　$x_j=FT(a,b).\quad(a,b)\notin g_i$

f)　　　End For

g)　　End Map

h)　　MapReduce.Reduce（ g_i,ε ）

i)　　　Spilt g_i to block b_i

j)　　　For each b_i,b_j in g_i

k)　　　　Calculate $FCI(b_i,b_j)$

l)　　　　Save key-value $<(x,y),FCI(x,y)>$

m)　　　End For

n)　　　While（ $<(x,y),FCI(x,y)>$ ）

o)　　　　if（ $FCI(x,y)<\varepsilon$ ）

p)　　　　　Delete data block where index = x && y

q)　　　End While

r)　　End Reduce

s)　End For

t)　Return $batch$

2.2　模型并行训练

在目前在大数据环境下的 DCNN 算法中，模型的并行训练需要将特征图与卷积核分散到不同的计算节点进行运算，但在构建并行卷积运算的过程中，算法难以筛除分散在各节点的冗余卷积核，导致在大数据环境下，传统 DCNN 算法无法解决卷积层运算速度慢的问题。为了解决此问题，本文提出 IM-PMTS 策略，该策略主要包含两个步骤：a)卷积核剪枝：设计马氏距离中心值 $MDCV$ (Mahalanobis distance center value)，通过求解 $MDCV$ 值来寻找与网络模型中卷积核线性相关的向量，并计算此向量到各个卷积核之间的距离 $dist$，通过设定阈值 α，裁剪 $dist<\alpha$ 的卷积核来减少网络模型中冗余参数；b)并行 Im2col 卷积：利用 Im2col 算法将特征图映射成矩阵，将矩阵与对应卷积核存储键值对，分发到各计算节点进行矩阵运算来加快卷积层的运算，得到运算卷积层运算结果，并将结果存入 HDFS 中。

1)卷积核剪枝

为了减少卷积神经网络中冗余卷积核所产生的无效计算，设计马氏距离中心值 $MDCV$ 筛除当前卷积层中冗余卷积核，进而加速卷积层运算，其具体过程为：首先，各节点计算卷积层所有的卷积核 $X_1,X_2,\ldots,X_n$ 的协方差矩阵 S 和均值 μ，构建目标函数 $MDCV$ 的目标函数 $f(x)$；接着，计算 $f(x)$ 在其驻点 x_k 处的二阶泰勒展开 $f(x)=x_k+\nabla f(x_k)(x-x_k)+\frac{1}{2}(x-x_k)^T\nabla^2 f(x_k)(x-x_k)$，若当前二阶导数非奇异，则下一个迭代点为 $x_{k+1}=x_k+\nabla^2 f(x_k)^{-1}\nabla f(x_k)$，若当前二阶导数奇异，先求解 $\nabla^2 f(x_k)d=-\nabla f(x_k)$ 确定搜索方向 d_k，在确定下一个迭代点 $x_{k+1}=x_k+d_k$，直至找到最优 $MDCV$ 值；最后，计算卷积层中所有卷积核到 $MDCV$ 值的距离 $dist$，并设定阈值 α，裁剪 $dist<\alpha$ 的卷积核完成卷积核剪枝过程。

定理 3　马氏距离中心值 $MDCV$．已知 $X_1,X_2,\ldots,X_n$ 表示网络模型中的卷积核，S 表示所有卷积核的协方差矩阵，μ 表示所有卷积核的均值。马氏距离中心值 $MDCV$ 的计算公式如下：

$$MDCV=\boldsymbol{x}^*=min\sum_{x\in R^n}\sqrt{(\boldsymbol{x}-\mu)^{\mathrm{T}}S^{-1}(\boldsymbol{x}-\mu)} \tag{7}$$

证明　$MDCV$ 是特征向量 $\boldsymbol{x}^*$ 到特征向量组 $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n$ 的最小距离，设 S 为向量组 $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n$ 的协方差矩阵，μ 为向量组的均值，其中引入协方差矩阵 S 来排除变量之间的相关性的干扰，当特征向量 $\boldsymbol{x}\to MDCV$ 值时，特征向量 $\boldsymbol{x}$ 就越容易被特征向量组替代，当 $\boldsymbol{x}=MDCV$，$\boldsymbol{x}$ 与 $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n$ 线性相关，故 $MDCV$ 值为表示特征向量 $\boldsymbol{x}^*$ 到特征向量组 $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n$ 的最小距离。证毕。

2)并行 Im2col 卷积

在完成卷积核剪枝后，便可结合 MapReduce 计算框架

实现 Im2col 卷积的并行运算，其具体过程为：首先，通过 Im2col 方法把输入特征图 M_i 映射为卷积计算矩阵 I_i，并将每张映射矩阵 I_i 与对应的卷积核存储键值对 $<I_i,K_z>$；接着，调用 Map()函数，将键值对中的矩阵 I_i 与对应卷积核的一维向量做矩阵相乘运算，得到卷积中间结果；最后，调用 Reduce()函数合并同一条数据的特征图，获得最终输出特征图 NM_i。

算法 2　模型并行训练算法

输入：卷积核 K ,输入特征图 M ,超参 α 。

输出：　输出特征图 NM 。

a) RunMapRedece (K,M,α)
b) For each k_z in K do
c) 　MapReduce.Map (k_z,α)
d) 　　Calculate $MDCV$
e) 　　While ($|k_z - MDCV| \le \alpha$)
f) 　　　Delete k_z
g) 　　End while
h) 　End Map
i) 　MapReduce.Reduce (k_z,M)
j) 　　I_i = Im2col (k_z,M)
k) 　　Save key-value $<I_i,K_z>$
l) 　　NM = $I_i \times K_z$
m) 　End Reduce
n) End For
o) Return NM

2.3　参数并行更新

目前大数据下的并行 DCNN 算法中，分布式集群中各节点首先进行正向传播获得各卷积层结果，并将结果统一传递至 Master 节点进行聚合，再通过反向传播，采用随机梯度下降法或批梯度下降法进行参数的更新。然而，在实现梯度下降的过程中，异常节点的训练数据会使得反向传播过程中的损失函数收敛震荡，进而导致损失函数收敛性差。为解决此问题，提出 IM-BGDS 策略，该策略主要包含两个步骤：a)梯度构建。提出损失均值权重 $LAW(g_i)$ (Loss Average Weight)来排除异常节点的训练数据对批梯度的影响，并设计损失求和梯度 $LSG(T)$ (Loss Sum Gradient)来构建批数据平均梯度，解决了损失函数收敛性差的问题。b)参数并行更新。在得到批数据的平均梯度后，结合 MapReduce 计算框架和反向传播的误差传导公式来并行化地计算误差，实现参数的并行更新。

1)梯度构建

为了排除异常节点的训练数据对批梯度的影响，设计损失均值权重 $LAW(g_i)$ 和损失求和梯度 $LSG(T)$ 来解决损失函数收敛性差的问题，其具体过程为：首先，根据损失函数公式计算批数据损失函数的均值，并计算批数据的损失函数与此均值的差，得到损失均值权重 $LAW(g_i)$，将结果映射为键值对 $<g_i,LAW(g_i)>$ 存入 HDFS 中；接着，计算批数据中每条数据 g_i 的损失函数的对当前参数 δ_z 的偏导 $\nabla J_{\delta i}$，同样将结果映射为键值对 $<g_i,\nabla J_{\delta i}>$ 存入 HDFS 中；最后，以 g_i 为索引遍历键值对 $<g_i,LAW(g_i)>$ 和 $<g_i,\nabla J_{\delta i}>$，构造批数据平均梯度 $LSG(T)$，获得当前参数的批梯度。

定理 4　损失均值权重 $LAW(g_i)$。已知 g_i 表示批数据中的一条数据，$J(\omega,b)_i$ 表示数据 g_i 损失函数值，$batch_size$ 表示批数据大小，$LAD(g_i)$ 为数据 g_i 的损失函数值与损失函数值均值之差的绝对值。损失均值权重 $LAW(g_i)$ 的计算公式如下：

$$LAW(g_i)=\begin{cases}1 & LAD(g_i)<\tau \\ 0 & LAD(g_i)\ge\tau\end{cases} \tag{8}$$

其中：

$$LAD(g_i)=\left|\frac{\sum_{i=1}^{batch_size} J(\omega,b)_i}{batch_size}-J(\omega,b)_i\right| \tag{9}$$

证明　$LAW(g_i)$ 是数据 g_i 的损失函数值的权重指标，设 $batch_size$ 为批数据大小，τ 为衡量 $LAD(g_i)$ 的阈值，当 $LAD(g_i)<\tau$ 时，则当前数据 g_i 的损失函数值属于常规值，故令 $LAW(g_i)=1$ 将其保留；当 $LAD(g_i)\ge\tau$ 时，则当前数据 g_i 的损失函数值属于异常值，故令 $LAW(g_i)=0$。证毕。

定理 5　损失求和梯度 $LSG(T)$。已知 T 表示批中所有数据，∇J_{xi} 表示数据 g_i 的损失函数对于参数 x 的梯度，$batch_size$ 表示批数据大小。损失求和梯度 $LSG(T)$ 的计算公式如下：

$$LSG(T)=\frac{\sum_{i=1}^{batch_size}\nabla J_{xi}\times LAW(g_i)}{batch_size} \tag{10}$$

证明　$LSG(T)$ 是批数据 $batch$ 的平均梯度，设 ∇J_{xi} 为数据 g_i 的损失函数对于参数 x 的梯度，$batch_size$ 为批数据大小，当 $LIW(g_i)=1$ 时，数据 g_i 的梯度 ∇J_{xi} 朝着最优方向下降；当 $LIW(g_i)=0$ 时，数据 g_i 的梯度 ∇J_{xi} 与最优方向偏差较大，不计入 $LSG(T)$ 梯度之中。证毕。

2)参数并行更新

在获得批数据平均梯度后，使用误差反向传播算法并行化的对误差项参数进行更新，得到参数并行更新后的网络模型，参数并行更新过程具体为：首先，计算第 $l-1$ 层卷积核 W_k^{l-1} 所有参数的梯度 $\sum_{i=1}^{k}LSG(T)^{l-1}$，并将结果映射为键值对 $<W_k^{l-1},\sum_{i=1}^{k}LSG(T)^{l-1}>$ 存入 HDFS 中；接着，计算网络模型中卷积核 W_k^{l-1} 参数的改变量 ΔW_k^{l-1}，以此更新第 $l-1$ 层卷积核的网络参数；最后，通过 HDFS 将更新后参数同步至所有计算节点，并进行下一步更新，直至网络模型中所有参数更新完成。

算法 3　参数并行更新算法

输入：批数据 $batch$,模型参数 W ,超参 τ

输出：　更新后网络模型参数 W

a) RunMapRedece ($batch,W,\tau$)
b) For each g_i in $batch$ do
c) 　Calculate $LSG(T)$ by using $LAW(g_i)$
d) 　MapReduce.Map ($W_k^{l-1},LSG(T)^{l-1}$)
e) End For
f) For Each $<W_k^{l-1},\sum_{i=1}^{k}LSG(T)^{l-1}>$ do
g) 　MapReduce.Reduce($W_k^{l-1},LSG(T)^{l-1}$)
h) 　　Calculate ΔW_k^l by using $LSG(T)^{l-1}$
i) 　　Update W_k^l by using ΔW_k^l
j) 　End Reduce
k) End For
l) Return $\sum W$

2.4　IA-PDCNNOA 算法的并行化流程

IA-PDCNNOA 算法的并行化流程具体实现步骤如下：

a)在特征并行提取阶段，输入原始数据集，启动一次 MapReduce 任务，按照数据类别划分为若干 *chunk*，依次将 *chunk* 中的数据输入到 *mapper* 节点中执行 MHO-PFES 策略，根据目标特征压缩原始数据集发送至 *reducer* 节点，最后将 *reducer* 节点中的目标特征保存至 HDFS。

b)在模型并行训练阶段，读取 HDFS 中的数据并随机打乱，划分为若干 *batch*，启动一个新的 MapReduce 任务，依次将 *batch* 中的数据输入到 *mapper* 节点执行 IM-PMTS 策略，进行卷积、ReLU、池化等操作，得到下一阶段特征图，最终得到输出的预测值存入 HDFS 中。

c)在参数并行更新阶段，Master 节点上读取上一阶段 *batch* 的输出的预测值，执行 IM-BGDS 策略，根据反向传播公式求全连接层、卷积层参数批梯度，经过多次循环步骤 b)c)，求解损失函数最小值，得到最终训练的网络模型。

IA-PDCNNOA 算法的并行化流程如图 1 所示。

2.5　算法时间复杂度分析

IA-PDCNNOA 算法的时间复杂度主要由特征并行提取、

模型并行训练和参数并行更新三个步骤构成。各部分具体时间复杂度计算如下：

a)特征并行提取阶段是结合 MapReduce 计算框架的并行单元运算结构，其时间复杂度主要分为(a) 并行架构下的数据特征提取；(b)主节点对数据的特征筛选两个部分，设样本数为 n，集群节点数为 k，拉普拉斯最大迭代次数为 m，算法在数据特征提取阶段利用 a,b 两个滑动窗口对数据进行遍历，并计算目标窗口 a 的为交叉零点，其时间复杂度为 $O(m\cdot n\cdot\log n/k)$；算法在特征筛选阶段计算任意两个数据切片相似度的时间复杂度为 $O(n^2)$，则特征并行提取阶段的时间复杂度为

$$T_1 = O(m\cdot n\cdot\log n/k + n^2) \tag{11}$$

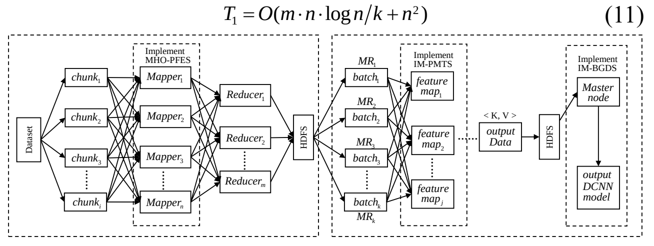


图 1 IA-PDCNNOA 算法并行化流程

Fig. 1 Parallelization flowchart of IA-PDCNNOA algorithm

b) 在模型并行训练阶段，算法提出(a) $MDCV$ 值的求解；(b) 卷积并行运算两个部分，设集群节点数为 k，模型中卷积核数量为 p，卷积核尺寸为 s，样本数为 n，算法在 $MDCV$ 值求解的过程需要求解卷积核的标准差，并寻找最大线性相关卷积核，其时间复杂度为 $O(p\cdot s^2/k)$，经过上个阶段的数据处理，算法在卷积并行运算时只需要做矩阵乘法运算，其时间复杂度为 $O(n^2)$，则模型并行训练的时间复杂度为

$$T_2 = O(p\cdot s^2/k + n^2) \tag{12}$$

c) 在参数并行更新阶段，算法提出了批数据的梯度构建，设集群节点数为 k，模型全连接输入的尺寸为 c，批数据梯度构建的时间复杂度为 $O(k\cdot c^2)$，所以，参数并行更新的时间复杂度为

$$T_3 = O(k\cdot c^2) \tag{13}$$

综上，本文提出的 IA-PDCNNOA 算法的时间复杂度为 $T_{IA-PDCNNOA} = T_1 + T_2 + T_3 = O((m\cdot n\cdot\log n + p\cdot s^2)/k + n^2 + k\cdot c^2)$.

对于 FCNN[14]算法，该算法首先通过将稀疏视图转换为全视图，再构建数据重建和处理技术并行化进行高精度模型训练，因此 FCNN 时间复杂度为

$$T_{FCNN} = O(d\cdot n\cdot\log n\cdot s^2 + n^2) \tag{14}$$

其中 s 为卷积核尺寸，d 为算法迭代次数。

对于 SSOCNN[15]算法，该算法设计了连续内存地址读取的单跨步情况下的 im2col 算法加速方法，并利用通用矩阵乘法对列向量和卷积核进行卷积运算，因此 SSOCNN 时间复杂度为

$$T_{SSOCNN} = O(a\cdot n^2\cdot\log n\cdot k/k^2) \tag{15}$$

其中 a 为单跨步数。

对于 MR-FPDCNN [16]算法，该算法将信息共享搜索策略与萤火虫算法相结合来寻找网络模型最优参数，并通过 MapReduce 并行训练网络模型，因此 MR-FPDCNN 时间复杂度为

$$T_{MR-FPDCNN} = O((n\cdot\log n + p\cdot n^3)/k) \tag{16}$$

其中 p 为特征图剪枝数量。

由算法理论分析可得 IA-PDCNNOA、FCNN、SSOCNN 以及 MR-FPDCNN 算法的时间复杂度，在大数据环境下，n 的基数远大于其他指标，可知 $(m\cdot n\cdot\log n + p\cdot s^2)/k + n^2 + k\cdot c^2 < d\cdot n\cdot\log n\cdot s^2 + n^2 < a\cdot n^2\cdot\log n\cdot k/k^2 < (n\cdot\log n + p\cdot n^3)/k$，相比于 FCNN、SSOCNN 和 MR-FPDCNN 算法，本文提出的 IA-PDCNNOA 算法在大数据环境下有着更为理想的时间复杂度。

3 实验结果以及分析

3.1 实验环境

为了验证 IA-PDCNNOA 算法的性能表现，本文设计了相关实验。实验硬件包含一台 Master 机和七台 Slaver 机组成，所有节点的 CPU 都为 AMD Ryzen 7 3800X，内存 32G，GPU 为 NVIDIA RTX2080Ti，通过 1000Mb/s 的以太网相连。实验的编程环境为 python3.8，TensorFlow 2.3，JDK 1.8，Apache Hadoop 3.3，Windows 10 Enterprise 2016 LTSB，节点配置如表 1 所示。

表 1 实验中节点的配置

Tab. 1 Configuration of nodes in the experiment

Node Type	Node Name	IP Configuration
Master	master	192.168.111.1
Slave	slave_1~7	192.168.111.2~8

3.2 实验数据

实验采用 CIFAR10、CIFAR100、ImageNet 1K 和 CompCars 数据集：CIFAR10 数据集包含 10 个类别，由尺寸为 32*32 彩色图像组成，每个类有 6000 个图像，有 50000 条训练集和 10000 条测试集；CIFAR100 数据集包含 100 个类别，由尺寸为 32*32 彩色图像组成，每个类有 600 个图像，每个类各有 500 个训练集和 100 个测试集；ImageNet 是目前世界上最大的图像识别数据库，ImageNet 1K 包含 1000 个类别，120 多万条训练集和 50 000 条的验证集，通过边界填充保持图像长宽比，将图像调整为 224*224；CompCars 数据集共包含 208826 个车辆图片，共有 163 个汽车品牌的 1716 款车辆型号。数据集的具体信息如表 2 所示。

表 2 实验数据集

Tab. 2 Experimental dataset

图片	CIFAR10	CIFAR-100	ImageNet 1K	CompCars
数/条	60 000	60000	1281 167	208826
尺寸/像素	32*32	32*32	224*224	224*224
类别/类	10	100	1000	1716

3.3 实验准备

本文采用 ResNet50 作为算法的训练网络，ResNet50 作为神经网络中具有跨层连接的代表，能够很好的反映出卷积神经网络算法对模型的优化效果。接着为了减小频繁读取小文件的开销，将图像转换为灰度图，并通过 MapReduce 算法并行化地将数据集中的图片转为 TFRECORD 格式，完成实验数据准备。

3.4 评价指标

实验主要通过模型的加速比，$Top-1$ 准确率，浮点运算量 FLOPs(Floating Point Operations)和算法运行时间 4 个评价指标衡量算法性能，加速比和 $Top-1$ 准确率定义如下：

3.4.1 加速比

加速比是通过并行计算以降低总体的运行时间而获得的性能提升的数值化表示形式，加速比越大，算法并行程度越高，定义如下：

$$S_n = T_s/T_n \tag{17}$$

其中，T_s 为算法串行运行时间，T_n 为算法并行运行时间。

3.4.2 $Top-1$ 准确率

$Top-1$ 准确率是深度学习中评价模型预测错误率的重要指标，$Top-1$ 准确率越高，模型性能越好，定义如下：

$$ACC_{top-1} = T_b/N \tag{18}$$

其中，T_b 为所以验证集中正确标签在模型输出的最佳标记中的样本数，N 为样本总数。

3.5 算法可行性比较分析

为验证 IA-PDCNNOA 算法在大数据环境下的并行训练可

行性，采用算法的加速比来进行衡量，对 IA-PDCNNOA 算法在 CIFAR10、CIFAR100、ImageNet 1K 和 CompCars 数据集上进行测试。同时为确保实验结果的准确性，取各算法平均 10 次运行时长来计算加速比，作为最后实验结果。实验结果如图 2 所示。

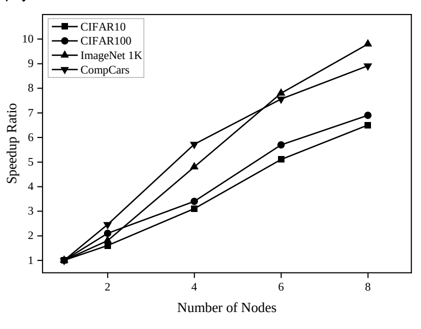


图 2　IA-PDCNNOA 算法在四个数据集的加速比

Fig. 2　Speedup ratio of IA-PDCNNOA algorithm in four datasets

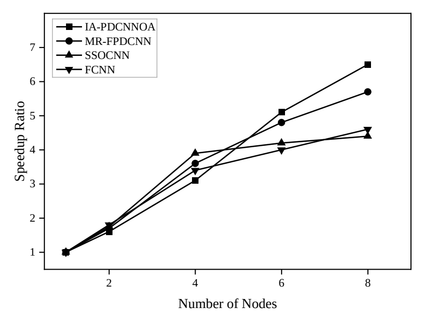
从图 2 可以看出，IA-PDCNNOA 算法随着节点数的增加，其加速比总体呈现上升趋势，且随着四个数据集规模的增加逐步增长。其中当节点数为 2 时，IA-PDCNNOA 算法在四个数据集上的加速比差异较小；当节点数为 4 时，算法相比于单节点的加速比分别增加了 2.205、2.417、3.824 和 4.735；当节点数为 8 时，IA-PDCNNOA 算法在各数据集上有了显著提升，分别达到了 6.861、6.876、9.828 和 8.915。这是由于 IA-PDCNNOA 算法设计了 IM-PMTS 策略，将等大小的图像块均匀分布至集群各计算节点，在减少节点通信时间的同时保证了数据的负载均衡，极大提升了算法的运行效率。此外算法还设计了 IM-BGDS 策略，排除了异常节点所产生的数据计算，避免了这类数据的读写与传输对系统资源的消耗，在一定程度上提升了算法的性能，随着数据规模的增大，这种提升的效果也逐渐明显。这也表明 IA-PDCNNOA 算法适用于大数据环境下，并行 DCNN 模型的训练。

3.6　算法性能实验分析比较

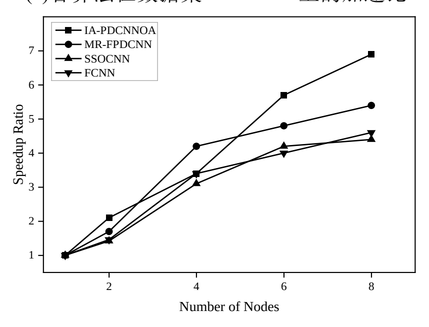
3.6.1 算法加速比实验分析

为验证 IA-PDCNNOA 算法在大数据环境中的并行化性能，本文基于 CIFAR10、CIFAR100、ImageNet 1K 和 CompCars 数据集，将加速比作为衡量指标，分别与 MR-FPDCNN、SSOCNN、FCNN，算法做比较。同时，为确保实验结果的准确性，取各算法平均 10 次运行时长来计算加速比，作为最后实验结果。实验结果如图 3 所示。

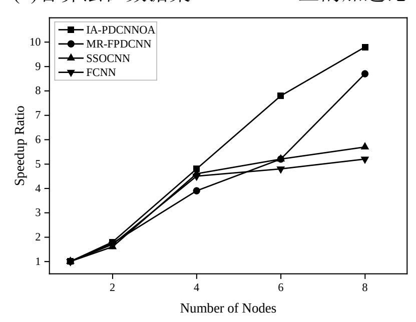
从图 3(a)(b)可以看出，在处理 CIFAR10、CIFAR100 这样规模相对较小的数据集时，各算法的加速比随着节点数的增加而缓慢增加，其中，当集群节点数为 4 时，IA-PDCNNOA 的加速比相比于并行化程度不高的 FCNN 和 SSOCNN 算法，分别低了 0.325、0.435 和 0.276、0.102；但在图 3(c)(d)中，算法处理 ImageNet 1K、CompCars 这样相对较大的数据集时，IA-PDCNNOA 算法的加速比增速较大，在集群节点数为 8 时分别达到了 9.804 和 8.912，相比 MR-FPDCNN、FCNN 和 SSOCNN 算法分别高出 1.148、4.173、4.652 和 0.965、2.678、2.094。产生这些结果的原因是：当 IA-PDCNNOA 算法在处理规模相对较小的数据集时，数据分布到各个计算节点会导致各节点间的通信时间开销快速增长，通过并行化运算获得的运行速度提升极为有限；当 IA-PDCNNOA 算法在处理规模相对较大的数据集时，因为其设计的 IM-PMTS 策略，通过提出马氏距离中心值 *MDCV* 对同层卷积核剪枝，减少了卷积层参数在网络通信中的开销，然后通过结合 MapReduce 和 Im2col 方法并行训练的方式加速卷积运算的过程，提高了卷积层运算速度，并提升了算法的加速比，实验表明，IA-PDCNNOA 算法并行化能力随着集群节点数的增多而显著增强，其适用于大数据集进行并行化处理，且具有较好的性能。



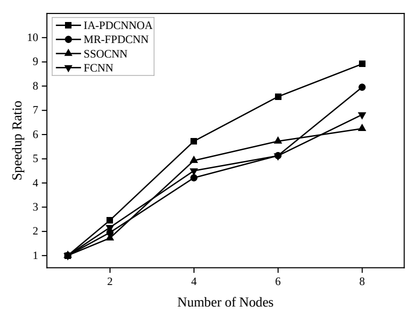
(a)各算法在数据集 CIFAR10 上的加速比



(b)各算法在数据集 CIFAR100 上的加速比



(c)各算法在数据集 ImageNet 1K 上的加速比



(d)各算法在数据集 CompCars 上的加速比

图 3　各算法在四个数据集的加速比

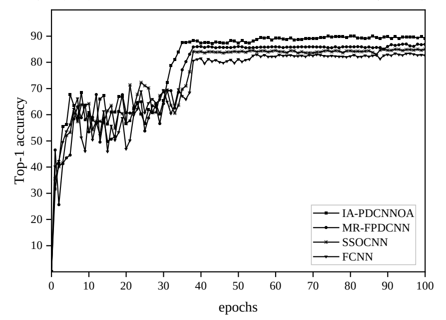
Fig. 3　Speedup ratio of each algorithm in four datasets

3.6.2 算法准确率实验分析

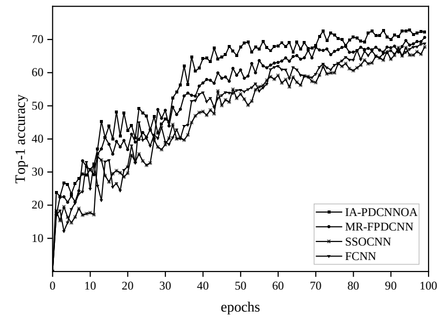
为了进一步验证 IA-PDCNNOA 算法的训练效果，使用 $Top-1$ 准确率作为衡量指标评价算法的训练效果，将 IA-PDCNNOA、MR-FPDCNN、SSOCNN 和 FCNN 分别在 CIFAR10、CIFAR100、ImageNet 1K 和 CompCars 数据集上进行训练，计算其 $Top-1$ 准确率作为实验结果，实验结果如图 4 所示。

从图 4(a)(b)可以看出，在处理 CIFAR10、CIFAR100 这样规模相对较小的数据集时，各算法的 $Top-1$ 准确率均能稳定在较高的数值，其中，IA-PDCNNOA 算法的 $Top-1$ 准确率最高，且较早的完成了收敛，达到了 89.72%和 72.31%，相比于 MR-FPDCNN、SSOCNN 和 FCNN 算法，高了 2.87%、4.62%、6.48%和 2.14%、4.57%、3.53%；但在图 4(c)(d)中，算法处理 ImageNet 1K、CompCars 相对较大的数据集时，各

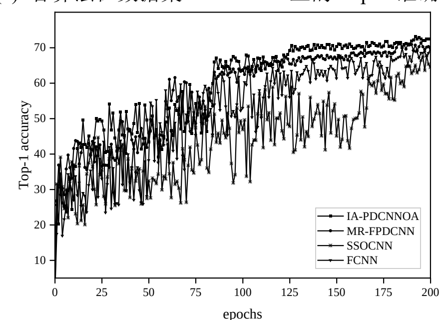
算法的 $Top-1$ 准确率和算法收敛情况有较大差异，其中，IA-PDCNNOA 算法的 $Top-1$ 准确率在四个并行化算法中最高，达到了 72.41%和 69.17%，相比于 MR-FPDCNN、SSOCNN 和 FCNN 算法，高了 2.31%、7.98%、2.85%和 2.81%、7.58%、4.84%，但其他三个算法均出现了不同程度难以收敛的情况。产生这些结果是： IA-PDCNNOA 算法提出 IM-BGDS 策略，其设计损失求和梯度 $LSG(T)$ 构建小批量数据梯度，并通过误差反向传播算法对参数并行更新，排除异常节点的训练数据对批梯度的影响，增强了 IA-PDCNNOA 算法的收敛性。因此可以得出，IA-PDCNNOA 相较于其他三个并行化算法有着较高的收敛速度和准确率，其适用于大数据集下的深度卷积神经网络的模型并行化训练。



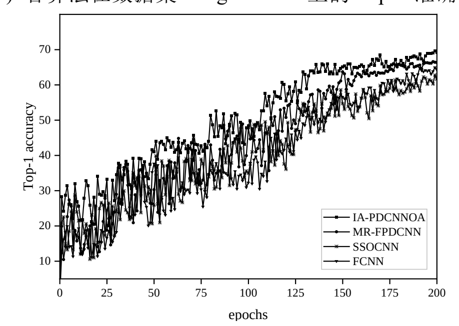
(a) 各算法在数据集 CIFAR10 上的 Top-1 准确率



(a) 各算法在数据集 CIFAR100 上的 Top-1 准确率



(c) 各算法在数据集 ImageNet 1K 上的 Top-1 准确率



(d) 各算法在数据集 CompCars 上的 Top-1 准确率

图 4　各算法在四个数据集上的 Top-1 准确率

Fig. 4　Top-1 accuracy of each algorithm on four datasets

3.6.3 算法运行时间和 FLOPs 实验分析

为验证 IA-PDCNNOA 算法在大数据环境中算法执行速度和模型优化效果，本文基于 CIFAR10、CIFAR100、ImageNet 1K 和 CompCars 数据集，分别计算 Baseline、IA-PDCNNOA、MR-FPDCNN、SSOCNN 和 FCNN 的运行时间和 FLOPs，其中 Baseline 为 ResNet50 模型在 1/8 数据负载量下的基准数据，实验结果如表 3 所示。

表 3　各算法在四个数据集上的运行时间和 FLOPs

Tab. 3　Running time and flops of each algorithm on four datasets

Dataset	Algorithm	Running time/s	FLOPs	Reduction of FLOPs
CIFAR10	Baseline	264	3.8×10^{9}	-
	MR-FPDCNN	186	1.97×10^{9}	48%
	SSOCNN	261	2.58×10^{9}	32%
	FCNN	242	2.39×10^{9}	37%
	IA-PDCNNOA	154	1.78×10^{9}	53%
CIFAR100	Baseline	427	3.8×10^{9}	-
	MR- FPDCNN	316	2.05×10^{9}	45%
	SSOCNN	368	2.87×10^{9}	25%
	FCNN	357	2.94×10^{9}	23%
	IA-PDCNNOA	281	1.91×10^{9}	50%
ImageNet 1K	Baseline	8.12×10^{4}	3.8×10^{9}	-
	MR- FPDCNN	6.23×10^{4}	2.62×10^{9}	31%
	SSOCNN	8.76×10^{4}	3.09×10^{9}	21%
	FCNN	1.02×10^{5}	2.81×10^{9}	26%
	IA-PDCNNOA	4.91×10^{4}	2.5×10^{9}	34%
CompCars	Baseline	6.72×10^{4}	3.8×10^{9}	-
	MR- FPDCNN	4.64×10^{4}	2.49×10^{9}	34%
	SSOCNN	7.72×10^{4}	2.98×10^{9}	22%
	FCNN	9.18×10^{4}	2.96×10^{9}	22%
	IA-PDCNNOA	3.59×10^{4}	2.41×10^{9}	37%

从表 3 可以看出，在处理 CIFAR10、CIFAR100 这样规模相对较小的数据集时，各算法运行时间没有较大的差距，但它们的浮点运算量均有不同程度的减少，其中， IA-PDCNNOA 的浮点运算量相比于 MR-FPDCNN、SSOCNN 和 FCNN 算法，分别减少了 5%、21%、16%和 5%、25%、27%；但在处理 ImageNet 1K、CompCars 这样较大的数据集时，IA-PDCNNOA 算法的运行时间和浮点运算量均优于其他三个算法，其中，IA-PDCNNOA 算法的运行时间相比于 MR-FPDCNN、SSOCNN 和 FCNN 算法快了 1.32×10^{4}s、3.85×10^{4}s、5.29×10^{4}s 和 1.05×10^{4}s、4.13×10^{4}s、5.59×10^{4}s，浮点运算量分别减少了 3%、13%、8%和 3%、15%、15%。对比四个算法在 CIFAR10、CIFAR100、ImageNet 1K 和 CompCars 数据集上的运行时间和浮点运算量的变化趋势，可以看出 IA-PDCNNOA 算法随着训练数据集的增大，其运行时间和浮点运算量的减少在与其他算法拉开了较大差距，产生这些结果是： IA-PDCNNOA 算法提出的 MHO-PFES 策略，其通过提出特征相关指数 $FCI(x,y)$，去除了数据中的冗余特征，并筛选数据的目标特征作为卷积神经网络的输入，减少了模型的浮点运算量，加快了算法的运行速度。因此可以得出，IA-PDCNNOA 优于 MR-FPDCNN、SSOCNN 和 FCNN，适用于大数据集下的 DCNN 模型并行化训练。

3.6.4 算法并行方式性能实验分析

为验证在大数据环境下，算法并行方式对模型构建时间影响，本文选取基于数据并行的 MR-FPDCNN 和基于模型并行的 SSOCNN 算法，与 IA-PDCNNOA 进行比较，算法 IA-PDCNNOA 在正向传播阶段是数据并行方式，反向传播阶段是模型并行方式。比较算法在 CIFAR10、CIFAR100、

ImageNet 1K 和 CompCars 数据集训练至模型准确率为 70%所需运行时间，实验结果如图 5 所示。

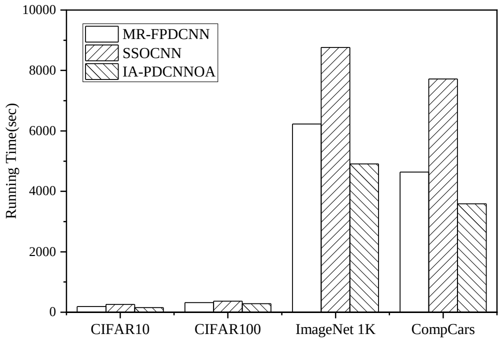


图 5　不同并行方式算法在四个数据集的运行时间

Fig. 5　Runtime of different parallel-mode algorithms on four datasets

从图 5 可以看出，算法在面对 CIFAR10、CIFAR100 这样的小规模数据集时训练时间相差不大，但在面对 ImageNet 1K、CompCars 这样的大数据环境下，IA-PDCNNOA 算法的运行时间相比于数据并行的 MR-FPDCNN 和模型并行的 SSOCNN 分别降低了 1322s、3837s 和 1049s、4127s，可以看出 IA-PDCNNOA 算法随着训练数据规模的增大，训练时间相比于数据并行的 MR-FPDCNN 和模型并行的 SSOCNN 出现了明显的优势。产生这些结果的原因是：对于数据并行算法 MR-FPDCNN，由于其将数据分散至不同节点单独训练，不同节点间的模型参数没有共享，导致需要花费更长时间训练才能达到目标准确率；对于模型并行算法 SSOCNN，其在卷积计算阶段各节点特征图合并使得算法承受了极大的通信开销，降低了算法的运行速度。相比于数据并行算法 MR-FPDCNN 和模型并行算法 SSOCNN，IA-PDCNNOA 在卷积计算的正向传播阶段，各节点分别计算 *batch* 中的特征图，免去了不同节点间的通信开销；在反向传播阶段，算法将计算结果构建批梯度训练模型参数，使得 IA-PDCNNOA 算法相比于 MR-FPDCNN 和 SSOCNN 算法，运行时间大幅度减少。实验表明，相比于数据并行算法 MR-FPDCNN 和模型并行算法 SSOCNN，IA-PDCNNOA 的混合式并行更适用于大规模的深度卷积神经网络的训练。

4　结束语

针对传统的深度卷积神经网络算法在大数据环境下的不足，本文提出一种基于 Im2col 算法的并行深度卷积神经网络优化算法 IA-PDCNNOA。首先，提出 MHO-PFES 策略，设计改进的非局部均值滤波器 $FT(a,b)$ 对输入数据进行滤波，并计算滤波数据的拉普拉斯方程 $h(x,y)$，寻找拉普拉斯方程的零交叉来提取数据特征，并提出特征相关指数 $FCI(x,y)$ 去除冗余数据，从而解决了数据冗余特征多的问题；然后，提出 IM-PMTS 策略，设计马氏距离中心值 $MDCV$ 寻找与网络模型中卷积核线性相关的向量，并以此对同层卷积核剪枝，然后通过结合 MapReduce 和 Im2col 方法并行训练的方式加速卷积运算的过程，提高了卷积层运算速度；最后，提出 IM-BGDS 策略，设计损失均值权重 $LAW(g_i)$ 来排除异常节点的训练数据对批梯度的影响，并设计损失求和梯度 $LSG(T)$，构建批数据平均梯度，并结合 MapReduce 计算框架和反向传播的误差传导公式对参数并行更新，排除异常节点的训练数据对批梯度的影响，解决了损失函数收敛性差的问题。为了验证 IA-PDCNNOA 算法的性能，本文在 ResNet50 网络上设计了相关实验，在 CIFAR10、CIFAR100、ImageNet 1K 和 CompCars 数据集上将 IA-PDCNNOA 算法分别于 MR-FPDCNN 算法、SSOCNN 算法和 FCNN 算法进行比较。最终的实验结果和实验分析均反映出于其他算法相比，IA-PDCNNOA 算法在处理大数据时具有相对较好的性能表现。虽然 IA-PDCNNOA 算法在深度卷积神经网络的模型训练方面取得进步，但该算法在预测准确率上依然存在一定的提升空间，算法的并行性能也有待加强，这将是今后的重点研究内容。

参考文献：

[1] 张珂，冯晓晗，郭玉荣，等. 图像分类的深度卷积神经网络模型综述[J]. 中国图像图形学报, 2021, 26 (10): 21. (Zhang Ke, Feng Xiaohan, Guo Yurong, *et al.* A review of deep convolution neural network models for image classification [J]. Journal of Image and Graphics, 2021, 26 (10): 21.)

[2] 杨真真，匡楠，范露，等. 基于卷积神经网络的图像分类算法综述[J]. 信号处理, 2018, 34 (12): 1474-1489. (Yang Zhenzhen, Kuang Nan, Fan Lu, *et al.* Review of Image Classification Algorithms Based on Convolutional Neural Networks [J]. Journal of Signal Processing, 2018, 34 (12): 1474-1489.)

[3] 朱方圆，马志强，陈艳，等. 语音识别中说话人自适应方法研究综述[J]. 计算机科学与探索, 2021, 15 (12): 15. (Zhu Fangyuan, Ma Zhiqiang, Chen Yan, *et al.* Survey of Speaker Adaptation Methods in Speech Recognition [J]. Journal of Frontiers of Computer Science and Technology, 2021, 15 (12): 15.)

[4] 罗会兰，袁璞，童康. 基于深度学习的显著性目标检测方法综述 [J]. 电子学报, 2021, 49 (7): 11. (Luo Huilan, Yuan Pu, Tong Kang. Review of the Methods for Salient Object Detection Based on Deep Learning [J]. Acta Electronica Sinica, 2021, 49 (7): 11.)

[5] 徐辉，祝玉华，甄彤，等. 深度神经网络图像语义分割方法综述 [J]. 计算机科学与探索, 2021, 15 (1): 13. (Xu Hui, Zhu Yuhua, Zhentong, *et al.* Survey of Image Semantic Segmentation Methods Based on Deep Neural Network [J]. Journal of Frontiers of Computer Science and Technology, 2021, 15 (1): 13.)

[6] 白子轶，毛懿荣，王瑞平. 视频人脸识别进展综述 [J]. 计算机科学, 2021, 48 (3): 10. (Bai Ziyi, Mao Yirong, Wang Ruiping. Survey on Video-based Face Recognition [J]. Computer Science, 2021, 48 (3): 10.)

[7] 朱向雷，王海弛，尤翰墨，等. 自动驾驶智能系统测试研究综述 [J]. 软件学报, 2021, 32 (7): 22. (Zhu Xianglei, Wang Haichi, Youhammer, *et al.* Survey on Testing of Intelligent Systems in Autonomous Vehicles [J]. Journal of Software, 2021, 32 (7): 22.)

[8] Fairuz, Amalina, Ibrahim, *et al.* Blending Big Data Analytics: Review on Challenges and a Recent Study [J]. IEEE Access, 2020, 8: 3629-3645.

[9] Vitor C. F. Gomes, Gilberto R. Queiroz, Karine R. Ferreira. An Overview of Platforms for Big Earth Observation Data Management and Analysis [J]. Remote Sensing, 2020, 12 (8): 1253-1253.

[10] 肖文，胡娟，周晓峰. 基于 MapReduce 计算模型的并行关联规则挖掘算法研究综述 [J]. 计算机应用研究, 2018, 35 (01): 13-23. (Xiao Wen, Hu Juan, Zhou Xiaofeng. Parallel association rules mining algorithm based on MapReduce: a survey [J]. Application Research of Computers, 2018, 35 (01): 13-23.)

[11] 金国栋，卞昊穹，陈跃国，等. HDFS 存储和优化技术研究综述 [J]. 软件学报, 2020, 31 (1): 137-161. (Jin Guodong, Bian haoqiong, Chen Yueguo, *et al.* Survey on Storage and Optimization Techniques of HDFS [J]. Journal of Software, 2020, 31 (1): 137-161.)

[12] Mahdi Mahmoud A. and Hosny Khalid M. and Elhenawy Ibrahim. Scalable Clustering Algorithms for Big Data: A Review [J]. IEEE ACCESS, 2021, 9: 80015-80027.

[13] Leung J, Chen M. Image Recognition with MapReduce Based Convolutional Neural Networks [C]// 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication

Conference (UEMCON) . IEEE, 2019, pp. 0119-0125.

[14] Takam C A, Samba O, Kouanou A T, *et al.* Spark Architecture for deep learning-based dose optimization in medical imaging [J]. Informatics in Medicine Unlocked, 2020, 19: 100335.

[15] Wang H, Ma C. An optimization of im2col, an important method of CNNs, based on continuous address access [C]// 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE) . 2021, pp. 314-320.

[16] 毛伊敏，张瑞朋，高波. 大数据下基于特征图的深度卷积神经网络 [J/OL]. 计算机工程与应用. 2022, 1-9. [2022-02-07]. http://kns. cnki. net/kcms/detail/11. 2127. TP. 20210413. 1143. 010. html. (Mao Yimin, Zhang ruipeng, Gao Bo. Deep convolutional neural network algorithm based on feature map in big data environment [J/OL]. Computer Engineering and Applications, 2022, 1-9. [2022-02-07]. http://kns. cnki. net/kcms/detail/11. 2127. TP. 20210413. 1143. 010. html.)

[17] Park J, Kang C K, Lee Y. Quantitative evaluation of the image quality using the fast nonlocal means denoising approach in diffusion-weighted magnetic resonance imaging with high b-value [J]. Journal of the Korean Physical Society, 2021, 78 (3): 244-250.

[18] 陈俊，何庆. 基于余弦相似度的改进蝴蝶优化算法 [J]. 计算机应用, 2021, 41 (09): 2668-2677. (Chen Jun, He Qing. Improved butterfly optimization algorithm based on cosine similarity [J]. Journal of Computer Applications, 2021, 41 (09): 2668-2677.)

[19] Ji Z, Zhang X, Wei Z, *et al.* A tile-fusion method for accelerating Winograd convolutions [J]. Neurocomputing, 2021, 460: 9-19.

[20] 王燕，亓祥惠，段亚西. 基于核函数与马氏距离的 FCM 图像分割算法 [J]. 计算机应用研究, 2020, 37 (02): 611-614+624. (Wang Yan, Qi Xianghui, Duan Yaxi. Image segmentation of FCM algorithm based on kernel function and Markov distance [J]. Application Research of Computers, 2020, 37 (02): 611-614+624.)